日　本　国　特　許　庁
JAPAN PATENT OFFICE

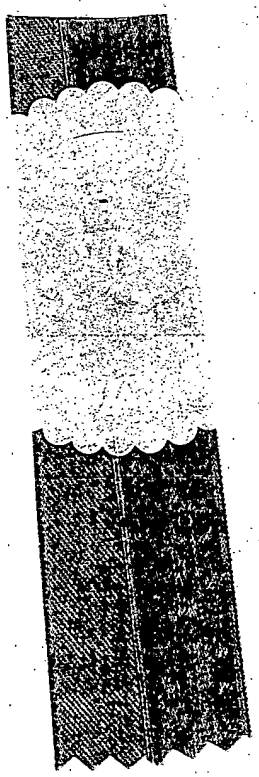
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　９月１１日

出願番号
Application Number: 特願２００１－２７４４３３

出願人
Applicant(s): 株式会社日立製作所



CERTIFIED COPY OF
PRIORITY DOCUMENT

２００１年１１月１６日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号　出証特２００１－３０９９７７

【書類名】　特許願

【整理番号】　K01014611A

【あて先】　特許庁長官

【国際特許分類】　G09C　1/00

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　渡辺　　大

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　古屋　聡一

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　宝木　和夫

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社日立製作所

【代理人】

【識別番号】　100075096

【弁理士】

【氏名又は名称】　作田　康夫

【先の出願に基づく優先権主張】

【出願番号】　特願2001- 13959

【出願日】　平成13年 1月23日

【先の出願に基づく優先権主張】

【出願番号】　特願2001-145783

【出願日】　平成13年 5月16日

【手数料の表示】

【予納台帳番号】　013088

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9902691

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　疑似乱数生成装置またはそれを用いた暗号復号処理装置

【特許請求の範囲】

【請求項１】ステート記憶部と、

バッファと、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するステート変換部と、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するバッファ変換部と、

クロックに応じて、前記ステート変換部出力を用いて前記ステート記憶部の内部状態を更新するステート記憶制御部と、

前記クロックに応じて、前記バッファ変換部出力を用いて前記バッファの内部状態を更新するバッファ制御部と、を備える疑似乱数生成装置であって、

前記ステート記憶部は3ブロック(ただし１ブロックはnビットからなる)の容量を備え、前記バッファは複数ブロックの容量を備え、

前記ステート変換部は、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを入力として用いる非線形変換部と、

前記変換結果のうち1ブロックデータを部分乱数列として出力する出力部と、を備える

ことを特徴とする疑似乱数生成装置。

【請求項２】請求項１記載の疑似乱数生成装置であって、

前記ステート変換部は、第1の演算部と第2の演算部とを備え、

前記第1の演算部は、前記ステート記憶部に記憶されている3つのブロックのうち、第1と第2の二つのブロックと、バッファに記憶されているブロックを入力として受け付ける入力部と、前記第1のブロックと前記バッファに記憶されているブロックを非線形変換し、nビットのデータを出力する第1の非線形変換部と、前記第1の非線形変換部出力と前記第2のブロックとを入力とし論理演算する第3の演算部と、前記第1のブロックと前記第3の演算部による演算結果を出力する出力

部と、を備え、

前記第2の演算部は、前記第1の演算部のいずれかの出力と前記ステート記憶部に記憶されている第3のブロックと、前記バッファに記憶されているブロックを入力として受け付ける入力部と、前記第1の演算部のいずれかの出力と前記バッファに記憶されているブロックを非線形変換し、nビットのデータを出力する第2の非線形変換部と、前記第2の非線形変換部出力と前記第3のブロックとを入力とし論理演算する第4の演算部と、前記第1の演算部のいずれかの出力と前記第4の演算部による演算結果を出力する出力部と、を備える

ことを特徴とする疑似乱数生成装置。

【請求項３】請求項2記載の疑似乱数生成装置であって、

前記ステート変換部は、さらに、転置部を備え、

前記転置部は、前記第3、第4の演算部の演算結果が、それぞれの前記演算部に入力されたブロックとは異なるブロックとして、前記ステート記憶部に記憶されるように転置する

ことを特徴とする疑似乱数生成装置。

【請求項４】請求項１記載の疑似乱数生成装置であって、

前記ステート変換部は、以下の処理を行う

$x_L \leftarrow a_H$;

$x_H \leftarrow a_M$ XOR $F(a_H, b_i)$

$x_M \leftarrow a_L$ XOR $G(x_H, b_j)$

(ただし、ステート記憶部の記憶内容の上位ブロックをa_H、中位ブロックをa_M、下位ブロックをa_L、前記バッファ記憶部の第iブロックをb_i、前記非線形変換部を$F(a, b)$、$G(a, b)$、データの代入を←、前記変換結果の上位ブロックをx_H、中位ブロックをx_M、下位ブロックをx_Lと表し、$i \neq j$とする)

ことを特徴とする疑似乱数生成装置。

【請求項５】請求項１記載の疑似乱数生成装置であって、

前記ステート変換部は、以下の処理を行う

$x_L \leftarrow a_M$;

$x_M \leftarrow a_H$ XOR $F(a_M, b_i)$

$x_H \leftarrow a_L$ XOR $G(a_M, b_j)$

(ただし、前記ステート記憶部の記憶内容の上位ブロックをa_H、中位ブロックをa_M、下位ブロックをa_L、前記バッファ記憶部の第jブロックをb_j、前記非線形変換部をF(a, b)、G(a, b)、データの代入を←、前記変換結果の上位ブロックをx_H、中位ブロックをx_M、下位ブロックをx_Lと表し、i≠jとする)

ことを特徴とする疑似乱数生成装置。

【請求項６】請求項１記載の疑似乱数生成装置であって、

1ブロックを64ビットで構成し、

前記非線形変換部は、入力されるブロックを8ビット単位に分けて非線形変換を行うSボックスをさらに備え、以下の処理を行う処理部を備える

$p \leftarrow a$ XOR b,

$t_i \leftarrow S[p_i]$ $(1 \leqq i \leqq 8)$;

$u_H \leftarrow t_1||t_2||t_3||t_4$;

$u_L \leftarrow t_5||t_6||t_7||t_8$;

$u_X \leftarrow u_X$ XOR $SHR_8(u_X)$, X={L, H};

$u_X \leftarrow u_X$ XOR $SHL_{16}(u_X)$, X={L, H};

$u_L \leftarrow u_H$ AND 0xf0f0f0f0;

$u_H \leftarrow u_L$ AND 0x0f0f0f0f;

out $\leftarrow u_H || u_L$;

(ただし、ステート記憶部からの入力をa、バッファからの入力をb、データの代入を←、$p=p_1||p_2||p_3||p_4||p_5||p_6||p_7||p_8$,$(1 \leqq i \leqq 8)$とし、Sボックス出力を、上位から$t_1$, t_2, t_3, t_4, t_5, t_6, t_7, t_8、またはS[x]、64ビット幅でのxビット分の右シフト、左シフトをそれぞれSHR_X, SHL_Xと表す)

ことを特徴とする疑似乱数生成装置。

【請求項７】請求項１記載の疑似乱数生成装置であって、

前記バッファの容量を32ブロックとし、前記バッファ変換部は、以下の処理を行う処理部を備える

前記バッファが出力する32ブロックのうち、上位から第25番目ブロックと第32番目ブロックとを除いたブロックをひとつ下位のブロックとして出力する；

第32番目ブロックの上位と下位を入れ替えたものと第25番目ブロックとの排他的論理和演算を行い、当該演算結果を第24番目ブロックとして出力する；

第32番目ブロックとステート記憶部の出力1ブロックとの排他的論理和演算を行い、当該演算結果を第1番目ブロックとして出力する

ことを特徴とする疑似乱数生成装置。

【請求項８】暗号復号処理装置であって、

暗号化対象となる平文データと同じ長さの疑似乱数列を生成する疑似乱数生成装置と、

生成された疑似乱数列と平文データとを排他的論理和演算することによって暗号文データを算出し、出力する演算部と、を備え、

前記疑似乱数生成装置は

ステート記憶部と、

バッファと、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するステート変換部と、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するバッファ変換部と、

クロックに応じて、前記ステート変換部出力を用いて前記ステート記憶部の内部状態を更新するステート記憶制御部と、

前記クロックに応じて、前記バッファ変換部出力を用いて前記バッファの内部状態を更新するバッファ制御部と、を備え、

前記ステート記憶部は3ブロック(ただし１ブロックはnビットからなる)の容量を備え、前記バッファは複数ブロックの容量を備え、

前記ステート変換部は、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いる非線形変換部と、

前記変換結果のうち1ブロックデータを部分乱数列として出力する出力部と、を備える

ことを特徴とする暗号復号処理装置。

【請求項９】暗号復号処理装置であって、

復号化対象となる暗号文データを生成する際に用いた乱数列を決定するための情報を用いて、暗号文データと同じ長さの疑似乱数列を生成する疑似乱数生成装置と、

生成された疑似乱数列と暗号文データとを排他的論理和演算することによって平文データを算出し、出力する演算部と、を備え、

前記疑似乱数生成装置は

ステート記憶部と、

バッファと、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するステート変換部と、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するバッファ変換部と、

クロックに応じて、前記ステート変換部出力を用いて前記ステート記憶部の内部状態を更新するステート記憶制御部と、

前記クロックに応じて、前記バッファ変換部出力を用いて前記バッファの内部状態を更新するバッファ制御部と、を備え、

前記ステート記憶部は3ブロック(ただし１ブロックはnビットからなる)の容量を備え、前記バッファは複数ブロックの容量を備え、

前記ステート変換部は、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いる非線形変換部と、

前記変換結果のうち1ブロックデータを部分乱数列として出力する出力部と、を備える

ことを特徴とする暗号復号処理装置。

【請求項１０】

記憶装置とプロセッサとを備えた計算機に、

ステート記憶部と、

バッファと、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するステート変換部と、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するバッファ変換部と、

所定のプログラムステップにおいて、前記ステート変換部出力を用いて前記ステート記憶部の内部状態を更新するステート記憶制御部と、

所定のプログラムステップにおいて、前記バッファ変換部出力を用いて前記バッファの内部状態を更新するバッファ制御部と、を実現し、疑似乱数を生成させるプログラムであって、

前記ステート記憶部は、3ブロック(ただし1ブロックはnビットからなる)の容量を備え、前記バッファは複数ブロックの容量を備え、

前記ステート変換部は、

前記バッファの記憶内容と前記ステート記憶部の記憶内容とを用いる非線形変換部と、

変換結果のうち1ブロックデータを部分乱数列として出力する出力部と、を備える

ことを特徴とする疑似乱数を生成させるプログラム。

【請求項１１】請求項1記載の疑似乱数生成装置であって、

前記ステート変換部は、以下の処理を行う

$X_H \leftarrow A_M$

$X_M \leftarrow A_L$ XOR $F(A_M, B_I)$

$X_L \leftarrow A_H$ XOR $G(A_M, B_J)$

(ただし、前記ステート記憶部の記憶内容の上位ブロックをA_H、中位ブロックをA_M、下位ブロックをA_L、前記バッファ記憶部の第IブロックをB_I、前記非線形変換部をF(A, B)、G(A, B)、データの入力を←、前記変換結果の上位ブロックをX_H、中位ブロックをX_M、下位ブロックをX_Lと表し、I≠Jとする)

ことを特徴とする疑似乱数生成装置。

【請求項１２】請求項1記載の疑似乱数生成装置であって、

1ブロックを64ビットで構成し、

前記非線形変換部は、入力されるブロックを8ビット単位に分けて非線形変換を行うSボックスと、Sボックスの出力を32ビット単位で線形変換するMDS行列をさらに備え、以下の処理を行う処理部を備える

$P \leftarrow A$ XOR B;

$T_I \leftarrow S[P_I]$ $(1 \leqq I \leqq 8)$;

$U_H \leftarrow MDS_1(T_1, T_2, T_3, T_4)$;

$U_L \leftarrow MDS_2(T_5, T_6, T_7, T_8)$;

$U_H = X_1 \,||\, X_2 \,||\, X_3 \,||\, X_4$

$U_L = X_5 \,||\, X_6 \,||\, X_7 \,||\, X_8$

$OUT \leftarrow X_5 \,||\, X_6 \,||\, X_3 \,||\, X_4 \,||\, X_1 \,||\, X_2 \,||\, X_7 \,||\, X_8$;

(ただし、ステート記憶部からの入力をA、バッファ記憶部からの入力をB、データの代入を←、$P = P_1 \,||\, P_2 \,||\, P_3 \,||\, P_4 \,||\, P_5 \,||\, P_6 \,||\, P_7 \,||\, P_8$ $(1 \leqq I \leqq 8)$とし、Sボックスの出力を上位から$T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$、またはS[X]、MDS行列による変換部を$MDS(T_a, T_b, T_c, T_d)$と表す)

ことを特徴とする疑似乱数生成装置。

【請求項１３】請求項1記載の疑似乱数生成装置であって、

前記バッファの容量を18ブロックとし、前記バッファ変換部は、以下の処理を行う処理部を備える

前記バッファからの入力18ブロックのうち、上位から第2番目ブロック、第12番目ブロック、第18番目ブロックを除いたブロックを一つ下位のブロックとして出力する；

第2番目ブロックと第7番目ブロックとの排他的論理和演算を行い、当該演算結果を第3番目ブロックとして出力する；

第15番目ブロックの上位1/2ブロックと下位1/2ブロックを入れ替えたブロックと第12番目ブロックとの排他的論理和演算を行い、当該演算結果を第13番目ブロックとして出力する；

第18番目ブロックとステート記憶部の1ブロックとの排他的論理和演算を行い、当該演算結果を第1番目ブロックとして出力する

ことを特徴とする疑似乱数生成装置。

【請求項１４】請求項1記載の疑似乱数生成装置であって、

鍵情報を前記バッファ部の容量と同じ大きさのデータに伸長して前記バッファ部に入力する鍵変換部を備える

ことを特徴とする疑似乱数生成装置。

【請求項１５】請求項1記載の疑似乱数生成装置であって、

前記ステート記憶部は公開パラメータを用いる

ことを特徴とする疑似乱数生成装置。

【請求項１６】請求項1記載の疑似乱数生成装置であって、

1ブロックを64ビットで構成し、

前記非線形変換部は、入力されるブロックを8ビット単位に分けて非線形変換を行うSボックスと、Sボックスの出力を32ビット単位で線形変換するMDS行列と、64ビットの定数をさらに備え、以下の処理を行う処理部を備える

$P \leftarrow A \text{ XOR } B;$

$T_I \leftarrow S[P_I]$ $(1 \leqq I \leqq 8);$

$U_H \leftarrow MDS_1(T_1, T_2, T_3, T_4);$

$U_L \leftarrow MDS_2(T_5, T_6, T_7, T_8);$

$U_H = X_1 \,||\, X_2 \,||\, X_3 \,||\, X_4;$

$U_L = X_5 \,||\, X_6 \,||\, X_7 \,||\, X_8;$

$Z \leftarrow X_5 \,||\, X_6 \,||\, X_3 \,||\, X_4 \,||\, X_1 \,||\, X_2 \,||\, X_7 \,||\, X_8;$

$OUT \leftarrow Z \text{ XOR } C;$

(ただし、ステート記憶部からの入力をA、バッファ記憶部からの入力をB、データの代入を←、$P = P_1 \,||\, P_2 \,||\, P_3 \,||\, P_4 \,||\, P_5 \,||\, P_6 \,||\, P_7 \,||\, P_8$ $(1 \leqq I \leqq 8)$とし、Sボックスの出力を上位からT_1, T_2, T_3, T_4, T_5, T_6, T_7,T_8、またはS[X]、MDS行列による変換部を$MDS(T_a, T_b, T_c, T_d)$、上記定数をCと表す)

ことを特徴とする疑似乱数生成装置。

【請求項１７】請求項16記載の疑似乱数生成装置において、

上記定数Cを8ビットからなるブロックに分けたとき、少なくとも一つのブロックは他と異なる値である

ことを特徴とする疑似乱数生成装置。

【請求項１８】請求項1記載の疑似乱数生成装置であって、

前記バッファの容量を16ブロックとし、前記バッファ変換部は、以下の処理を行う処理部を備える

前記バッファからの入力16ブロックのうち、上位から第4番目ブロック、第10番目ブロック、第16番目ブロックを除いたブロックを一つ下位のブロックとして出力する；

第4番目ブロックと第8番目ブロックとの排他的論理和演算を行い、当該演算結果を第5番目ブロックとして出力する；

第14番目ブロックの上位1/2ブロックと下位1/2ブロックを入れ替えたブロックと第10番目ブロックとの排他的論理和演算を行い、当該演算結果を第11番目ブロックとして出力する；

第16番目ブロックとステート記憶部の1ブロックとの排他的論理和演算を行い、当該演算結果を第1番目ブロックとして出力する

ことを特徴とする疑似乱数生成装置。

【請求項１９】請求項1記載の疑似乱数生成装置であって、

鍵情報と乱数列番号とを入力とする鍵変換部と、前記鍵変換部を制御する制御部とを備え、

前記鍵変換部制御部は、

前記鍵情報を前記バッファ部の容量と同じ大きさのデータに伸長して前記バッファ部に入力し、前記鍵情報を前記ステート部の容量と同じ大きさのデータに伸長して前記ステート部に入力するよう、前記鍵変換部を制御し、　前記伸長されて前記ステート部に入力された鍵情報と、前記乱数列番号とを用いて、さらに、前記ステート部のデータを更新するように、前記ステート変換部と前記鍵変換部とを制御する

ことを特徴とする疑似乱数生成装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、実用的な乱数列を生成する技術とその応用技術に関する。

【０００２】

【従来の技術】

公開鍵暗号技術を利用した署名生成または、秘密通信を行う際の鍵の生成、ストリーム暗号(stream cipher)技術などにおいて、乱数の必要性は高い。

しかし、これらの場面において真性乱数を用いようとすることは非現実的であり，実際には疑似乱数生成方法またはそれを用いた装置により生成される疑似乱数(以下、単に乱数ともいう)が用いられる。

【０００３】

暗号での使用に関して疑似乱数に要求される条件として、予測不可能性や、乱数を決定するための初期値を生成した乱数から導出できないこと、という安全性に関する性質がある。

さらに疑似乱数生成方法または生成装置が実用に耐えるためにはソフトウェア実装またはハードウェア実装において高速な処理が求められる。

さらに、ハードウェア実装する場合の必要ゲート数や、ソフトウェア実装した場合のステップ数や実行時の必要メモリ領域などが小さい、といった実装コストの面からも効率的である必要がある。

汎用的な暗号アルゴリズムとして、これらの評価項目全てにおいて欠点のないものが好ましい。

【０００４】

【発明が解決しようとする課題】

現在知られているアルゴリズムは、性能または実装面でソフトウェア、ハードウェアいずれか一方に適したものが多い。

たとえば、特にソフトウェア実装に適したアルゴリズムは、ハードウェア実装時のことが考慮されていないために回路規模が大きくなる。

【０００５】

上記のようなソフトウェア処理に適したアルゴリズムは、たとえば、以下に開示されている。

文献1：Daemen, J., Clapp, C., “Fast Hashing and Stream Encryption with PANAMA,” Fast Software Encryption, 5th International Workshop, Procee

dings, LNCS1372, 61-74, Springer-Verlag, 1998.

文献2：米国特許 5,454,039号。

【０００６】

ハードウェア処理に適したアルゴリズムを用いたものの例としては、ハードウェア実装規模が小さく高速な処理も実現できる、線形フィードバックシフトレジスタ(linear feedback shift register (LFSR))にもとづいた疑似乱数生成装置があるが、ソフトウェア実装した場合の処理速度が充分ではない。

LFSRにもとづいた疑似乱数生成装置は、たとえば、以下に開示されている。

文献3：B. Schneier, "Applied Cryptography," John Wiley & Sons, Inc., 1996.pp. 369～428。

【０００７】

ソフトウェア、ハードウェアいずれにおいても実用的な実装が可能な疑似乱数生成方法として、ブロック暗号(block cipher)技術を利用したOFBモード(OFB mode)、カウンタモード(counter mode)などが知られているが、その乱数生成速度はブロック暗号の処理速度と同じであり、一般に専用の疑似乱数生成装置に比べて処理速度が充分ではない。

ブロック暗号技術を利用した技術については，文献3，pp. 203～206に記載されている。また、その処理速度は，文献2と以下に示す文献4に記述される暗号の処理速度から評価することができる。

【０００８】

文献4：B. Schneier, D. Whiting, "Fast Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor," Fast Software Encryption, 4th International Workshop, FSE'97, Haifa, Israel, January 1997, Proceedings, Lecture Notes in Computer Science Vol. 1267, Springer-Verlag, pp.242-259, 1998。

【０００９】

暗号技術の応用分野の広まりとともに、ハードウェア、ソフトウェアいずれにおいても上記各条件を満たす、実用になる自由度と柔軟性を備えた疑似乱数生成技術が望まれている。

また、より高い安全性を備えた疑似乱数生成技術が望まれている。

【００１０】

【課題を解決するための手段】

本発明は、安全性が高く、ソフトウェアでも高速処理が可能であり、かつ、ハードウェアでも高速かつ回路規模の小さい実装が可能な、疑似乱数生成方法または装置を提供する。

本発明は、また、上記疑似乱数生成技術を用いた暗号化装置または復号化装置を提供する。

本発明は、その一態様において、

上記文献1が開示するアルゴリズムを実行する疑似乱数生成装置が備えるバッファ(大きなメモリ領域)とステート(より小さなメモリ領域)のうち、ステートについて、(i)データ処理の単位長をn(たとえば32、64、128、256ビット)としたとき、ステートの大きさが3以上(好ましくは3)×nビットであり、(ii)クロック制御により、時刻tから時刻t+1へのステートの状態変更を行うためのステート変換部(ステート変換関数)が、(nビット入力、nビット出力)の非線形関数Fを2回、または異なる二つの非線形関数F，Gを一回ずつ使うという構成を提供する。

【００１１】

他の態様によれば、本発明による疑似乱数生成装置は、ステート記憶部と、バッファと、バッファの記憶内容とステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するステート変換部と、バッファの記憶内容とステート記憶部の記憶内容とを用いた変換を行い変換結果を出力するバッファ変換部と、クロックに応じて、ステート変換部出力を用いてステート記憶部の内部状態を更新するステート記憶制御部と、クロックに応じて、バッファ変換部出力を用いてバッファの内部状態を更新するバッファ制御部と、を備え、ステート記憶部は3ブロック(ただし１ブロックはnビットからなる)の容量を備え、バッファは複数ブロックの容量を備え、ステート変換部は、バッファの記憶内容とステート記憶部の記憶内容とを用いる非線形変換部と、変換結果のうち1ブロックデータを部分乱数列として出力する出力部と、を備える。

また、上記構成は、記憶装置とプロセッサとを備えた計算機上にプログラムに

よって実現されるものであってもよく、ステート記憶部とバッファ部は、該プログラムの所定のステップにおいてその内部状態が更新される。

【００１２】

また、ステート変換部は、第1の演算部と第2の演算部とを備え、第1の演算部は、ステート記憶部に記憶されている3つのブロックのうち、第1と第2の二つのブロックと、バッファに記憶されているブロックを入力として受け付ける入力部と、第1のブロックとバッファに記憶されているブロックを非線形変換し、nビットのデータを出力する第1の非線形変換部と、第1の非線形変換部出力と第2のブロックとを入力とし論理演算する第3の演算部と、第1のブロックと第3の演算部による演算結果を出力する出力部と、を備え、

第2の演算部は、第1の演算部のいずれかの出力とステート記憶部に記憶されている第3のブロックと、バッファに記憶されているブロックを入力として受け付ける入力部と、第1の演算部のいずれかの出力とバッファに記憶されているブロックを非線形変換し、nビットのデータを出力する第2の非線形変換部と、第2の非線形変換部出力と第3のブロックとを入力とし論理演算する第4の演算部と、第1の演算部のいずれかの出力と第4の演算部による演算結果を出力する出力部と、を備える。

さらに、上記ステート変換部は、第3、第4の演算部の演算結果が、それぞれの演算部に入力されたブロックとは異なるブロックとして、ステート記憶部に記憶されるように転置する転置部を備える。

【００１３】

上記構成は、ステートの大きさを3ブロック以上×nビットとすることにより、並列処理を可能としている。さらに、ステートの大きさを3ブロック×nビットとすることにより、ハードウェア実装時の回路規模を小さくすることが可能になる。

すなわち、上記構成は、(a) 安全性の評価が簡単でありながら、より高い安全性を保証でき、(b) ソフトウェア実装やハードウェア実装において高速な、(c) 必要メモリ領域やハードウェア実装時のゲート数が少なく、実装コストが安い疑似乱数装置の実現を可能とする。

【００１４】

【発明の実施の形態】

(用語の説明)

疑似乱数生成装置(pseudorandom number generator)：乱数列(random number sequence)を決定するための初期値(initial value)を与えて、疑似乱数列(pseudorandom number sequence)を生成する装置。

疑似乱数(pseudorandom number)：有限、もしくは無限のビット列であり、どのような方法でも真正乱数と区別することができないもの。

真正乱数：無限ビット列であり、任意の連続する部分列が与えられても、次の1ビットを推定することができないような列のこと。

共通鍵暗号(symmetric-key encryption)：暗号化処理と復号化処理に同じ鍵を用いる暗号化技術。

鍵：暗号化処理の際に用いる秘匿パラメータ。

平文(plaintext)：暗号化処理前、または復号化処理後のデータであり、ディジタル化されたマルチメディアデータも含まれる。

暗号文：暗号化処理されたデータ。

ブロック暗号：入力データを一定長のデータごとに区切り(区切られた一定長データをブロックという)、鍵と共に攪拌処理を行うことで暗号化処理(encryption)または復号化処理(decryption)を行う暗号技術。

ストリーム暗号：疑似乱数生成装置に乱数列を決定する情報を与えて乱数列を生成し、この乱数列と平文を攪拌することで暗号文を生成する暗号技術。

非線形変換：状態遷移関数のうち、線形変換でないもの。

Sボックス(S箱)：3～10ビット程度の置換表。高い非線形性と攪拌性を伴った変換を表参照で行えるうえ、簡単な構成による実現が可能であることから、暗号の実装に多く用いられる。

最大分離距離符号(MDS)行列：線形変換であって、入出力データの非0となる元の個数の和の最小値(分岐数)が最大となる行列。n×nのMDS行列は分岐数がn+1であることが知られている。

【００１５】

（実施例）

本発明の実施形態を図を用いて説明する。各図の説明に用いている記号「XOR」はビットごとの排他的論理和(exclusive or)を表す。

図2に示すように、本実施例の乱数生成装置(200)は、1ブロックを64ビットとしたとき64ビット×3ブロックの記憶領域を備えるステート記憶部(以下、ステートという)(201)および、64ビット×32ブロックの記憶領域を備えるバッファ(202)、ステート(201)の内部状態を更新するステート変換部(203)、バッファの内容を更新するバッファ変換部(204)、バッファ入力を切り替えるスイッチ(207)とそれを制御する制御部(214)で構成される。

ディジタル回路またはプログラム構成上、32の倍数を処理単位とすることが望ましく、暗号学的な安全性の観点から、ステートとバッファの内部状態数は大きい方がよい。好ましい本実施例においては、処理単位である１ブロックを64ビットとし、ステートの大きさを3ブロック、バッファの大きさを32ブロックとしている。これにより処理を並列化することと回路規模を小さくすることが可能になる。

【００１６】

本実施例をハードウェアにて構成した場合、トリガとなるクロック信号(210)を受信すると、ステート(201)は、ステート変換部(203)の値を新しいステート値として記憶する。バッファも同様にクロック信号(210)を受信すると、バッファ変換部(204)の値を新しい値として記憶する。制御回路(214)はクロック信号(210)を受信すると、内部カウンタにて計数し、その値に応じてスイッチ(207)が入力(212)とステート(201)出力のいずれを選択するかを切り替える。バッファ変換部(204)は、スイッチ(207)の切り替えに応じて決定される入力値(64ビット)と現在のバッファの値から新たなバッファ(202)の内容を決定する。

【００１７】

図5は、図2のバッファ変換部(204)の構成をより詳細に例示するものである。

バッファ変換部(204)は、バッファ(202)から入力された32ブロックのうち、上位から数えて25ブロック目と32ブロック目を除いたブロックをひとつ下位のブロックとして出力する。また、第25ブロックと、第32ブロックの上位と下位を入れ

替えたものとの排他的論理和演算を行った後一つ下位のブロックとして出力する。また、第32ブロックと、スイッチ(207)の出力との排他的論理和演算を行った後第1ブロックとして出力する。

【００１８】

図1のフローチャートにもとづいて図2の動作を説明する。

本実施例では、乱数生成装置(200)は、図1の処理ステップ102からステップ108を行う。また、1クロック毎に64ビットのビット列(部分乱数列)を生成する。

ステップ102から104において、内部状態の初期化、乱数生成のためのセットアップを行う。

ステップ102：ステート(201)、バッファ(202)の内容、および制御回路(214)内部のメモリをリセット入力(211)によりリセットする。例えば、すべてのビット値を0にする。

ステップ103：乱数列を決定する情報として、鍵情報と乱数列番号とを入力(212)する。入力に上位64ビットの鍵情報を与えて、クロック信号(210)を入力する。次に入力に下位64ビットの鍵情報を与えて、クロック信号(210)を入力する。続けて同様に、乱数列番号も上位、下位64ビットに分割し、2クロックで入力する。これらのクロック処理にはバッファ(202)やステート(201)は前述のとおりに動作する。さらに出力回数N(213)を入力する。本実施例では、生成する乱数列を決定するための鍵情報と乱数列番号、さらに出力回数をそれぞれ128ビットの数値とする。

ステップ104：初期状態を生成するために、128クロック処理を行い、ステート(201)、バッファ(202)の内容を攪拌する。制御回路(214)は、このステップ以降ではスイッチ(207)がステート出力を選択するように、制御する。本実施例においては、例えば、ステート出力の上位64ビットa_Hがスイッチ(207)に入力される。

【００１９】

以降のステップにおいて部分乱数列を繰り返し生成する。

ステップ105：N=0ならば終了(ステップ109)するが、そうでないならステップ106に進む。

ステップ106：ステート変換部(203)は変換処理を行い、部分乱数列64ビットを出力する。

ステップ107：バッファとステートに、クロック信号とステート変換部(203)とバッファ変換部(202)による変換処理結果とを与え、内容を更新する。

ステップ108：N=N-1として、ステップ105に戻る。

上記ステップ105からステップ108の繰り返しにより生成される一つ以上の部分乱数列を、例えばビット連結することにより所望の疑似乱数列を得る。

本実施例の構成をソフトウェア処理で実現する場合は、一つの部分乱数列を得る上記ステップ105からステップ108までの一連の操作をラウンドという。

【００２０】

図11は、バッファ変換部(204)の他の構成を例示するものである。このとき、協同するバッファ(202)の記憶領域の構成は64ビット×18ブロックとする。

バッファ変換部(204)は、バッファ(202)から入力された18ブロックのうち、上位から数えて第2、12、18ブロック目を除いたブロックを一つ下位のブロックとして出力する。また、第2ブロックと第7ブロックを排他的論理和した演算結果を第3ブロックとして出力する。また、第15ブロックの上位1/2ブロックと下位1/2ブロックを入れ替えたブロックと第12ブロックを排他的論理和した演算結果を第13ブロックとして出力する。さらに、第18ブロックと、スイッチ(207)の出力とを排他的論理和した演算結果を第1ブロックとして出力する。

図11に例示したバッファ変換部の構成は、図5の構成に比べて変換が複雑であり、バッファ内部をより強く攪拌できるので、暗号学的強度が増すという効果がある。

【００２１】

図12は、図11のバッファ変換部を用いた場合の乱数生成装置の構成を例示するものである。図12の構成における、図1のフローチャートのステップ103、104の動作を説明する。

ステップ103：秘匿パラメータとして鍵情報(1201)と、公開パラメータとして乱数列番号(1202)を入力する。本実施形態では、鍵情報、乱数列番号はいずれも128ビット(2ブロック)とする。ステート(201)には、下位2ブロックに乱数列番号

128ビットを入力する。

ステップ104：初期状態を生成する。

【００２２】

鍵変換部を用いて変換した鍵情報をバッファに入力する。鍵変換部は次のように変換を行う。

$K_H \,||\, K_L$ = 鍵情報(128ビット)

$Y_{2I} \leftarrow K_L >>> 7I \ (0 \leqq I < 9)$

$Y_{21+I} \leftarrow K_H <<<7(I+1) \ (0 \leqq I < 9)$

バッファへは鍵変換部の出力を上位から順に入力する。

$B_0 \,||\, B_1 \,||\, \ldots \,||\, B_{17}$ = バッファ値(64ビット×18ブロック)

$B_I \leftarrow Y_I \ (0 \leqq I < 18)$

本実施例で、例えば、ブロック暗号の段関数をステート変換部(203)として使用することができる(文献5参照)。ブロック暗号の段関数は、1段ではステート(201)の内部状態を十分に攪拌することはできない。

しかし、バッファ変換部(204)を用いてバッファの内部状態を動的に更新することにより、充分な攪拌が行えるので、ブロック暗号と同程度以上の安全性を保持することができる。上記のようにして実現した疑似乱数生成装置は、ブロック暗号の使用方法とみなすことができる。

【００２３】

図13は、バッファ変換部(204)の他の構成を例示するものである。このとき、協同するバッファ(202)の記憶領域の構成は64ビット×16ブロックとする。

バッファ変換部(204)は、バッファ(202)から入力された16ブロックのうち、上位から数えて第4、10、16ブロック目(すなわち、B_3、B_9、B_{15})を除いたブロックを一つ下位のブロックとして出力する。また、第4ブロック(B_3)と第8ブロック(B_7)を排他的論理和した演算結果を第5ブロック(すなわちY_4)として出力する。また、第14ブロック(B_{13})の上位1/2ブロックと下位1/2ブロックを入れ替えたブロックと第10ブロック(B_9)を排他的論理和した演算結果を第11ブロック (Y_{10})として出力する。さらに、第16ブロックと、スイッチ(207)の出力とを排他的論理和した演算結果を第1ブロック(Y_0)として出力する。

図13に例示したバッファ変換部の構成は、ブロックの数が2のベキであり、バッファ変換部をソフトウェアで実現した場合に、状態の更新が高速に行えるという利点を持つ。

【００２４】

図14は、図13のバッファ変換部を用いた場合の乱数生成装置(1400)の構成を例示するものである。図14の乱数生成装置が実行する、図1のフローチャートで行われる初期化を説明する。ただし、図1のフローチャートのステップ103，104として、図15のステップ1601～1604が実行される。

ステップ1601：秘匿パラメータとして鍵情報(1502)を鍵変換部(1504)に入力する。鍵情報は128ビットとする。鍵情報(1502)をK、C’ を64ビットの定数、64ビット単位の右巡回シフトを<<<、左巡回シフトを>>>と表すとき、鍵変換部(1504)は次のように変換を行い、鍵情報(1502)をステート変換部の処理単位と同じ大きさ(本実施例では192ビット)のデータAに伸長する。

【００２５】

すなわち、

$K_H \,||\, K_L = K$,

$A_H \leftarrow K_H$,

$A_M \leftarrow K_L$,

$A_L \leftarrow (K_H <<< 7)\ \mathrm{XOR}\ (K_L >>> 7)\ \mathrm{XOR}\ C'$

伸長したデータAをステート変換部(203)に入力する。

ステップ1602：バッファ部(202)への入力を、

$Y_0 \,||\, Y_1 \,||\, \ldots \,||\, Y_{15}$ = バッファ値(64ビット×16ブロック)

と表現し、上記データAをステート変換部(203)でi回変換したものをR_i、R_iの上位から64ビットごとにR_{Hi}，R_{Mi}，R_{Li}としたとき、下記のように、バッファ部(202)へR_{Hi}を下位から順に入力し、バッファ部(202)はこれらを保持する。

【００２６】

すなわち、

$Y_i \leftarrow R_{H(15-i)}$ $(0 \leqq i < 16)$

ステップ1603：公開パラメータである乱数列番号(1503)を入力する。本実施形

態では、乱数列番号(1503)は128ビットとする。乱数列番号Qは鍵変換部(1504)により次のような処理を経て192ビットに伸長され、ステート変換部(203)に入力される。

$Q_H \,||\, Q_L = Q$

$D_H \leftarrow R_{H16}$ XOR Q_H,

$D_M \leftarrow R_{M16}$ XOR Q_L,

$D_H \leftarrow R_{L16}$ XOR (Q_H <<< 7) XOR (Q_L >>> 7) XOR C' ,

$D = D_H \,||\, D_M \,||\, D_L$

乱数列番号Qが伸長されたデータDをステート変換部(203)で16回変換したものをEとし、Eをステート部(201)が保持する。

【００２７】

ステップ1604：ステップ1602と1603での処理結果が入力されたステート部、バッファ部の状態を16回更新する。

すなわち、バッファ部とステート部に、クロック信号とステート変換部(203)とバッファ変換部(202)による変換処理結果とを与え、内容を更新する。

【００２８】

本実施例の初期化方法によれば、初期化段階におけるデータの流れの把握が容易になる。したがって、安全性の評価を十分に行うことが可能となる。

【００２９】

図3、図4は、上記各実施例における、非線形なステート変換部(203)の演算部、転置部などの、実施例を示す図である。

図3の構成において、ステート(201)の変換は次のようにして行う。ただし、以下のアルゴリズムで、矢印←はデータの代入を、||は結合(concatenation，連結)を、それぞれ表す。ステート(201)の値をロードする。

$a_H \,||\, a_M \,||\, a_L$= ステート値(192ビット)

$x_L \leftarrow a_H$;

$x_H \leftarrow a_M$ XOR $F(a_H, b_i)$

$x_M \leftarrow a_L$ XOR $G(x_H, b_j)$(ただし$i \neq j$)

x_Mを部分乱数列として出力し、次のステート値として、$x_H \,||\, x_M \,||\, x_L$を出力

する。

【００３０】

この構造によれば、或るクロックt(またはあるラウンド)におけるG関数(402)への入力とクロックt+1(またはその次のラウンド)におけるF関数(401)への入力とが同じになるため、処理の並列化が可能になる。

また、或るクロックtにおける部分ステート値a_Lは、G関数(402)の出力と排他的論理和された後、クロックt+1においてF関数(401)の出力と排他的論理和されるため、F関数(401)とG関数(402)とは異なる変換を行う関数であることが望ましい。

【００３１】

図4はステート変換部(203)の他の構成を例示する図である。図4の構成において、ステート(201)の変換は次のようにして行う。ステート(201)の値をロードする。

$a_H \,||\, a_M \,||\, a_L$= ステート値(192ビット)

$x_L \leftarrow a_M$;

$x_M \leftarrow a_H$ XOR $F(a_M, b_i)$

$x_H \leftarrow a_L$ XOR $G(a_M, b_j)$ (ただし$i \neq j$)

x_Hを部分乱数列として出力し、次のステート値として、$x_H \,||\, x_M \,||\, x_L$を出力する。

図４に示す構造によれば、或るクロックt(またはあるラウンド)におけるF関数(401)への入力とG関数(402)への入力とが同じになり、処理の並列化が可能になる。

【００３２】

図9はステート変換部(203)の他の構成を例示する図である。図9の構成において、ステート(201)の変換は次のようにして行う。ステート(201)の値をロードする。

$A_H \,||\, A_M \,||\, A_L$ = ステート値(192ビット)

$X_H \leftarrow A_M$

$X_M \leftarrow A_L$ XOR $F(A_M, B_I)$

$X_L \leftarrow A_H$ XOR $G(A_M, B_J)$ （ただしI ≠ J）

各ラウンドにおいて、A_Hを部分乱数列として出力し、次のステート値としてX_H || X_M || X_Lを出力する。

図9に示す構造によれば、出力列として開示されたデータが変化しないラウンドが短いので、安全性が高まる。

【００３３】

上記図3、図4においてF関数とG関数に与えられるバッファからの入力(215)は、任意に選択可能である。たとえば、図6には示していないが、第5ブロック目と第17ブロック目を選択することが可能である。

上記図3，図4に示す構造以外の転置方法を採用しても良い。

【００３４】

上記図3、図4で用いたF関数(またはG関数)の構造を図6に例示する。

二つの入力データの排他的論理和演算結果を、8ビット毎に区切り、それぞれSボックス、S1(601)～S8(602)により非線形変換を行う。Sボックス S1(601)～S8(602)の8ビット出力に、さらに図6に示す変換処理を行い、出力outを生成する。

Sボックスとしては、たとえば

文献5：J. Daemen, V. Rijmen, "AES Proposal: Rijndael," The first AES Candidate Conference, available at http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip.のRijndaelで使われている以下の変換表を使うことが可能である。

【００３５】

```
S[256] =
{ 82,   9, 106, 213,  48,  54, 165,  56,
 191,  64, 163, 158, 129, 243, 215, 251,
 124, 227,  57, 130, 155,  47, 255, 135,
  52, 142,  67,  68, 196, 222, 233, 203,
  84, 123, 148,  50, 166, 194,  35,  61,
 238,  76, 149,  11,  66, 250, 195,  78,
   8,  46, 161, 102,  40, 217,  36, 178,
```

118, 91, 162, 73, 109, 139, 209, 37,
114, 248, 246, 100, 134, 104, 152, 22,
212, 164, 92, 204, 93, 101, 182, 146,
108, 112, 72, 80, 253, 237, 185, 218,
94, 21, 70, 87, 167, 141, 157, 132,
144, 216, 171, 0, 140, 188, 211, 10,
247, 228, 88, 5, 184, 179, 69, 6,
208, 44, 30, 143, 202, 63, 15, 2,
193, 175, 189, 3, 1, 19, 138, 107,
58, 145, 17, 65, 79, 103, 220, 234,
151, 242, 207, 206, 240, 180, 230, 115,
150, 172, 116, 34, 231, 173, 53, 133,
226, 249, 55, 232, 28, 117, 223, 110,
71, 241, 26, 113, 29, 41, 197, 137,
111, 183, 98, 14, 170, 24, 190, 27,
252, 86, 62, 75, 198, 210, 121, 32,
154, 219, 192, 254, 120, 205, 90, 244,
31, 221, 168, 51, 136, 7, 199, 49,
177, 18, 16, 89, 39, 128, 236, 95,
96, 81, 127, 169, 25, 181, 74, 13,
45, 229, 122, 159, 147, 201, 156, 239,
160, 224, 59, 77, 174, 42, 245, 176,
200, 235, 187, 60, 131, 83, 153, 97,
23, 43, 4, 126, 186, 119, 214, 38,
225, 105, 20, 99, 85, 33, 12, 125};

また、ステート記憶部からの入力をa、バッファからの入力をb、p=p1||p2||p3||p4||p5||p6||p7||p8,(1 ≦ i ≦ 8)とし、Sボックス S1(601)～S8(602)の8ビット出力を上位からそれぞれt1, t2, t3, t4, t5, t6, t7, t8としたとき、図6に示す変換処理は以下のようにも表せる。

【００３６】

下記の数式中で記号「S[x]」もSボックス S1(601)～S8(602)の8ビット出力を表し、「SHR_X, SHL_X」はそれぞれ64ビット幅でのxビット分の右シフト、左シフトを表す。

$p \leftarrow a$ XOR b;

$t_i \leftarrow S[p_i]$ $(1 \leqq i \leqq 8)$;

$u_H \leftarrow t_1||t_2||t_3||t_4$;

$u_L \leftarrow t_5||t_6||t_7||t_8$;

$u_X \leftarrow u_X$ XOR $SHR_8(u_X)$, $X=\{L, H\}$;

$u_X \leftarrow u_X$ XOR $SHL_{16}(u_X)$, $X=\{L, H\}$;

$u_L \leftarrow u_H$ AND 0xf0f0f0f0;

$u_H \leftarrow u_L$ AND 0x0f0f0f0f;

out $\leftarrow u_H || u_L$;

なお、図4に示す構造において、F関数(401)とG関数(402)とを同じ関数とし、さらに、バッファからの入力(211)を非線形変換の後に排他的論理和する構造にすれば、さらに処理の並列化による効果が高まる。

また、ステートとバッファの容量はそれぞれ3ブロック分、32ブロック分と小さいので、初期設定に必要なクロック数(ラウンド数)を少なくすることが可能である。さらに、ハードウェア化した場合のハードウェア規模を小さくすることができる。

【００３７】

F関数(401)(またはG関数(402))の他の構成を図10に示す。

二つの入力データの排他的論理和演算結果を、8ビット毎に区切り、それぞれSボックス1001により非線形変換を行う。次に、Sボックス1001の出力を32ビットごとに線形変換1002により線形変換を行う。さらに、32ビット出力それぞれ上位16ビットを交換し、出力OUTを生成する。

Sボックスとしては、たとえば文献5で使われている変換を用いることが可能である。また、32ビットの線形変換も、文献5で使われているものを用いることができる。

【００３８】

文献5で使われている線形変換は、最大距離分離符号行列と呼ばれ、入出力のデータをもっとも効率的に攪拌する線形変換として知られているものの一例である。このようにして構成したF関数は、図6に例示したF関数に比べ高い強度を持つ。このため、初期化に要するラウンド数を削減することができる。すなわち、バッファ(202)のステージ数を削減することができ、さらなるハードウェア規模の削減が可能である。

また、ステート記憶部からの入力をA、バッファ記憶部からの入力をB、データの代入を←、$P = P_1 \,||\, P_2 \,||\, P_3 \,||\, P_4 \,||\, P_5 \,||\, P_6 \,||\, P_7 \,||\, P_8$ ($1 \leqq I \leqq 8$)とし、Sボックスの出力を上位からT_1, T_2, T_3, T_4, T_5, T_6, T_7,T_8、またはS[X]、MDS行列による変換部をMDS(T_a, T_b, T_c, T_d)としたとき、図12に示す変換処理は以下のようにも表せる。

【００３９】

下記の数式中で記号「S[x]」もSボックス S(1001)の各8ビット出力を表す。

$P \leftarrow A \text{ XOR } B;$

$T_I \leftarrow S[P_I]$ $(1 \leqq I \leqq 8);$

$U_H \leftarrow MDS_1(T_1, T_2, T_3, T_4);$

$U_L \leftarrow MDS_2(T_5, T_6, T_7, T_8);$

$U_H = X_1 \,||\, X_2 \,||\, X_3 \,||\, X_4$

$U_L = X_5 \,||\, X_6 \,||\, X_7 \,||\, X_8$

$OUT \leftarrow X_5 \,||\, X_6 \,||\, X_3 \,||\, X_4 \,||\, X_1 \,||\, X_2 \,||\, X_7 \,||\, X_8;$

さらに、変換結果OUTを出力する前に、さらに定数Cとの排他的論理和演算を行っても良い。定数CはF関数、G関数で異なるものを用いても良い。

【００４０】

また、8ビット(1バイト)単位のブロックに分け、少なくとも一つのブロックが他と異なるようにすれば、バイト単位の変換を多用するF関数(401)にビット単位での変化をつけることができる。例えば、F関数、G関数に用いる定数をそれぞれC_F, C_Gとしたとき、

C_F = 0xBB67AE85 84CAA73B,

C_G = 0x3C6EF372 FE94F82B

とする。ただし、0xnnnnnnnnは数値が16進表記であることを表す。

【００４１】

本実施例の構成によれば、ステート変換部には、以上構成を例示した非線形変換部だけでなく、その他、暗号学的安全性と実装について十分に評価されたブロック暗号の非線形変換部、段関数を使用することができる。本実施例のように、処理単位を64ビットとすれば、ブロック暗号の非線形変換部、段関数の使用がより容易になる。また、ブロック暗号のOFBモードの3～5倍程度の処理速度を期待できる。

【００４２】

以上の説明における、64ビットという処理単位は一例であって、要求される仕様に応じて変更可能である。

また、本実施例を、演算処理装置(プロセッサ)と記憶装置を備える一般的な情報処理装置上にソフトウェアで実現する場合は、ステート、バッファを記憶装置または演算処理装置のレジスタを用いて実現し、その他各構成要素を、演算処理装置が記憶装置に格納されたプログラムを実行することにより実現する。各プログラムは、予め記憶装置に記憶されているか、または、情報処理装置が備える外部記憶媒体読取装置を介して記憶媒体から記憶装置にロードされる。または、通信装置を介して一旦ネットワークから記憶装置にダウンロードされてもよい。

また、いずれかの構成要素としてハードウェア化されたものを併用しても良い。

【００４３】

(応用例)

本実施例の一つの好ましい応用例は、たとえば、膨大なデータ処理を行う必要があるネットワークの基幹部や、高速なリアルタイム処理が必要なマルチメディアデータ処理システムである。以下、本実施例を利用したデータの暗号化および配信システムについて説明する。

【００４４】

図7は本実施例に用いる機器の概略図であり、図8は、図7のデータ送信装置(90

1)とデータ受信装置(907)中の暗号復号処理装置(904)を本実施例の疑似乱数生成装置を用いて構成した例である。

データの配信は、次のステップで行なう。

ステップ1：データ送信者とデータ受信者は、同じ乱数列を共有できるように、事前に鍵情報(1001)を秘密裏に共有し、乱数列番号(1002)も秘密裏である必要はないが共有する。これらの情報を共有するには、例えば、公開鍵暗号技術を用いた鍵配送方法を用いることができる。

ステップ2：データ送信者は、上記共有情報を与えた暗号復号処理装置(904)を用いて平文データ(903)を暗号化する。図8を使って暗号復号処理装置(904)の動作を説明する。

【００４５】

暗号復号処理装置(904)は、入力(1003)(この場合は平文データ(903))と同じ長さの疑似乱数列(1005)を生成する。疑似乱数列の生成は、本実施例による疑似乱数生成装置(200)に、上記鍵情報(1001)と乱数列番号(1002)と出力回数を与えることにより行う。生成された疑似乱数列(1005)とデータ(903)とを排他的論理和することによって出力(1004)として暗号文データ(1006)を算出し、出力する。

ステップ3：データ送信者は、暗号文データ(1006)をネットワークインターフェース装置(905)を用いて、ネットワーク(906)を通じてデータ受信者に送信する。

ステップ4：データ受信者は、ネットワークインターフェース装置(908)を介して受信した暗号文データ(1006)を、上記共有情報を与えた暗号処理装置(904)を用いて復号する。図8を使って暗号復号処理装置(904)の動作を説明する。

【００４６】

暗号復号処理装置(904)は、入力(1003)(この場合は暗号文データ(1006))と同じ長さの疑似乱数列(1005)を、暗号化時と同様に生成する。生成された疑似乱数列(1005)と暗号文データ(1006)とを排他的論理和することによって出力(1004)として復号文すなわち平文データ(1011)を算出し、出力する。

ステップ5：データ受信者は、復号化された平文データを再生装置(912)で再生する。

【００４７】

【発明の効果】

本発明によれば、ソフトウェア、ハードウェアいずれにおいても実装コストがかからず実用的な、さらに高速に動作可能な疑似乱数生成技術を提供できる。

【符号の説明】

200…疑似乱数生成装置、201…ステート、202…バッファ、203…ステート変換部、204…バッファ変換部、206…出力する乱数列、207…スイッチ、210…クロック入力、211…リセット入力、212…入力、213…平文の長さ情報Nの入力、214…制御回路、401…非線形変換F、601～604…8ビットの置換表、901…データ送信装置、902…記憶装置、903…データ、904…暗号復号処理装置、905…ネットワークインターフェース装置、906…ネットワーク、907…データ受信装置、908…ネットワークインターフェース装置、 910…記憶装置、912…データ再生装置、1001…鍵情報、1002…乱数列番号、1003…入力データ、1004…出力データ。

【図面の簡単な説明】

【図１】

実施例における疑似乱数生成装置の処理手順を表すフローチャートである。

【図２】

実施例における疑似乱数生成装置の概略構成図である。

【図３】

実施例における疑似乱数生成装置のステート変換部の構成を表す概略図である。

【図４】

ステート変換部の他の構成を表す概略図である。

【図５】

実施例における疑似乱数生成装置のバッファ変換部の構成を表す概略図である。

【図６】

ステート変換部で用いる非線形変換部の構成を表す概略図である。

【図７】

実施例における疑似乱数生成装置を応用したデータ配信システムの概略図である。

【図８】

実施例を応用したデータ送信装置の暗号復号処理装置の概略図である。

【図９】

ステート変換部のさらに他の構成を表す概略図である。

【図１０】

非線形変換部の他の構成を表す概略図である。

【図１１】

バッファ変換部の他の構成を表す概略図である。

【図１２】

図11に示すバッファ変換部を用いて構成する疑似乱数生成装置の概略構成図である。

【図１３】

バッファ変換部の他の構成を表す概略図である。

【図１４】

図13に示すバッファ変換部を用いて構成する疑似乱数生成装置の概略構成図である。

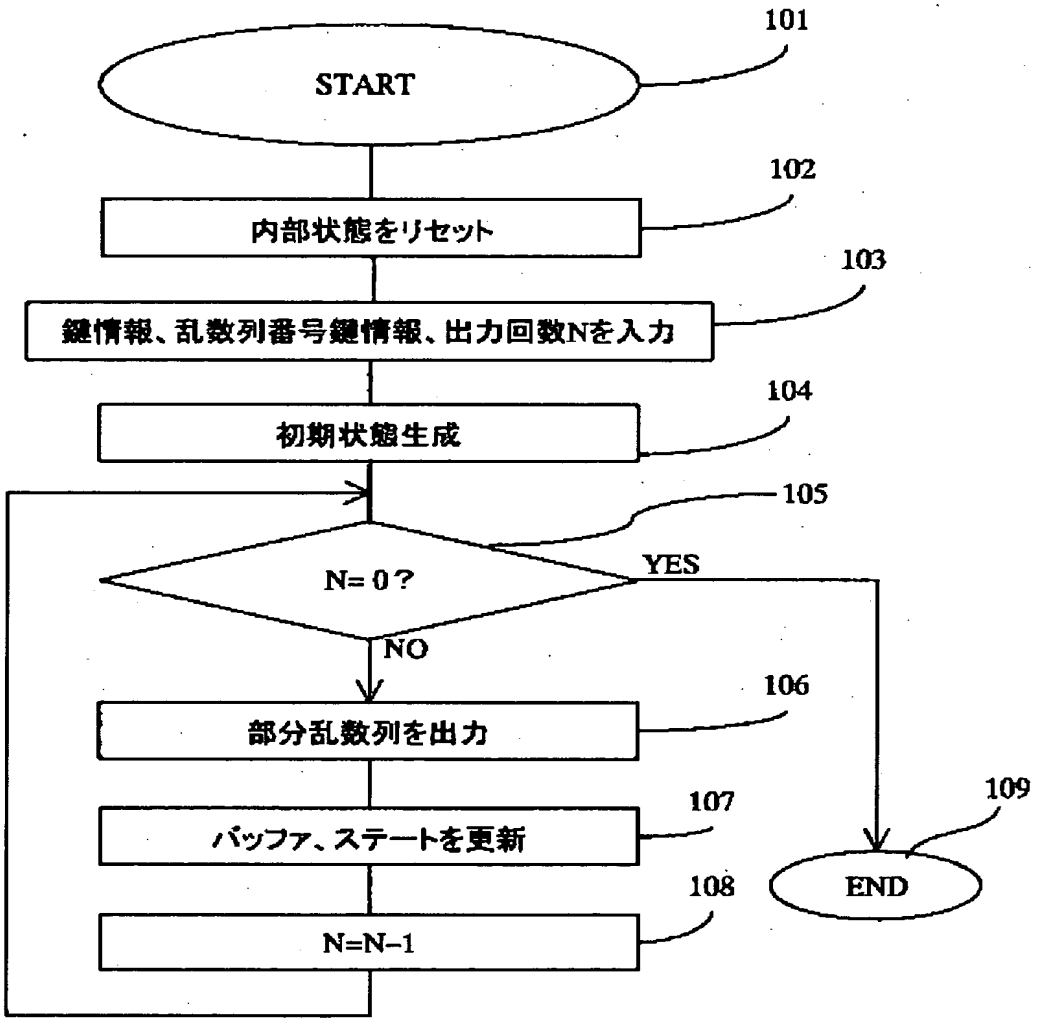
【図１５】

図14の疑似乱数生成装置が実行する初期化の動作を説明するフローチャートである。

【書類名】　図面

【図１】

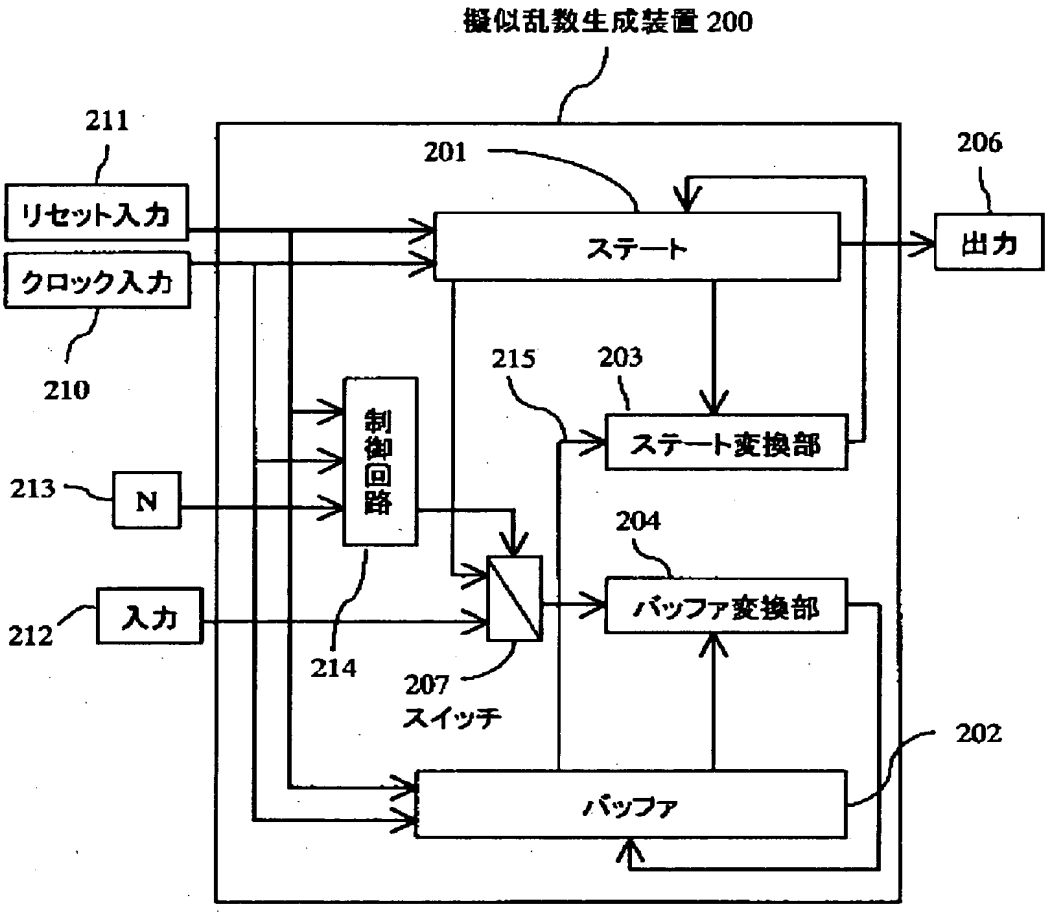
図1



出証特２００１－３０９９７７７

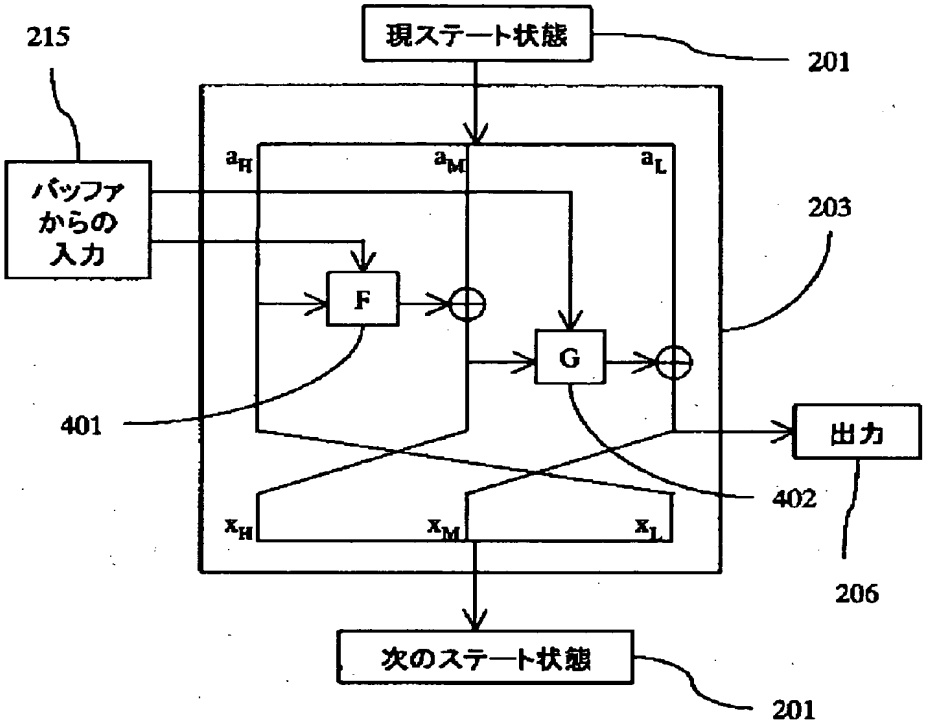
【図２】

図2


擬似乱数生成装置 200
211
リセット入力
クロック入力
210
201
ステート
206
出力
215
203
ステート変換部
制御回路
213
N
204
バッファ変換部
212
入力
214
207
スイッチ
202
バッファ

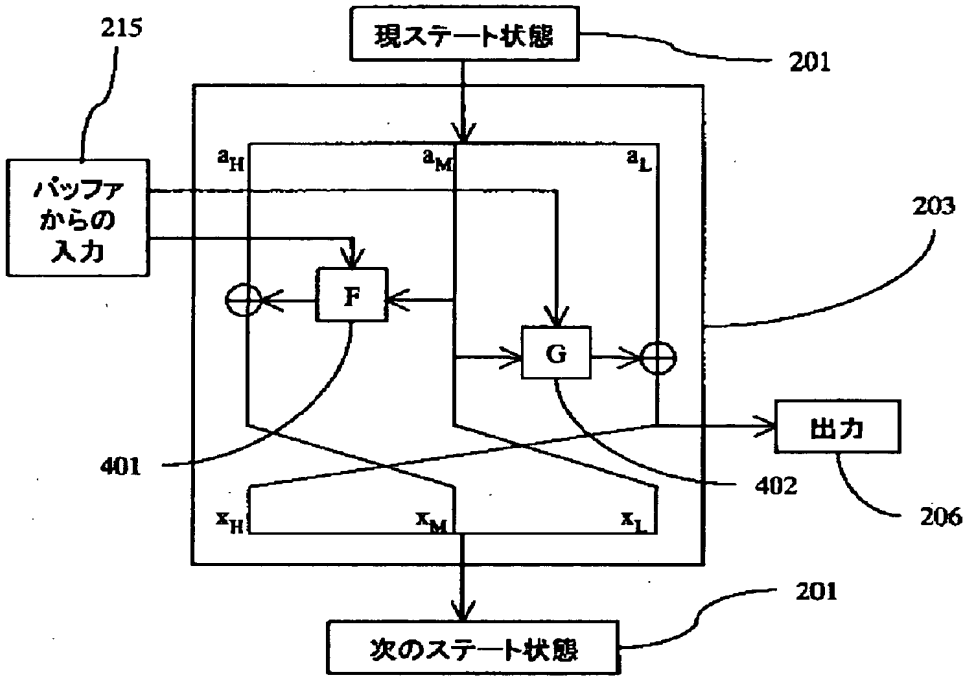
【図３】

図3


現ステート状態
201
215
バッファ
からの
入力
a_H
a_M
a_L
203
F
G
401
出力
402
x_H
x_M
x_L
206
次のステート状態
201

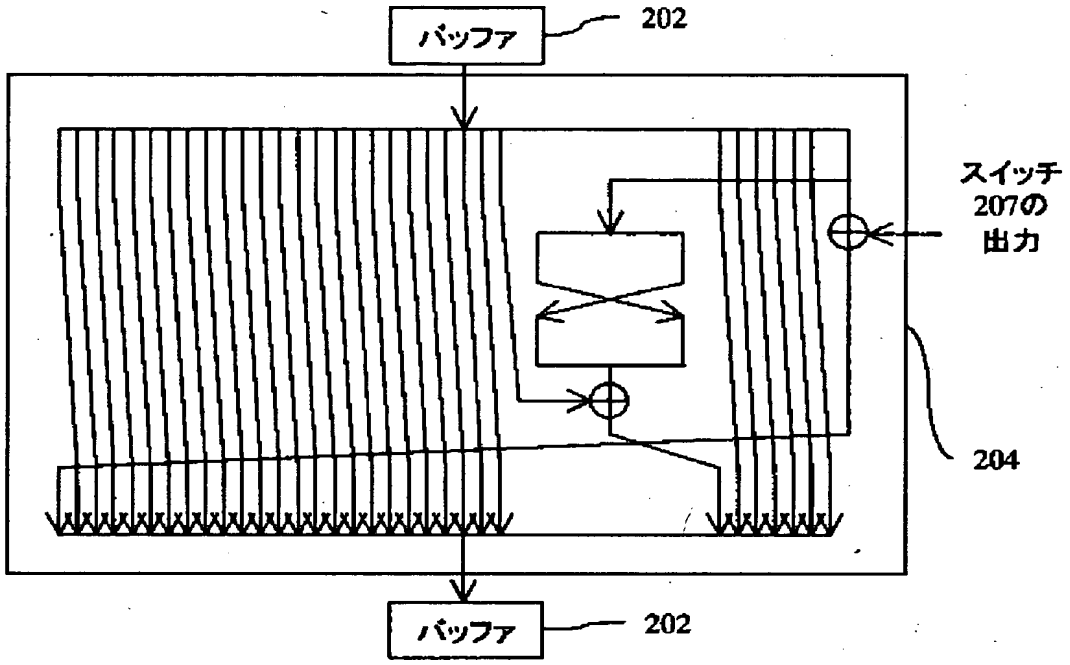
【図４】

図4


215
現ステート状態
201
パッファ
からの
入力
a_H
a_M
a_L
203
F
G
出力
401
402
206
x_H
x_M
x_L
201
次のステート状態

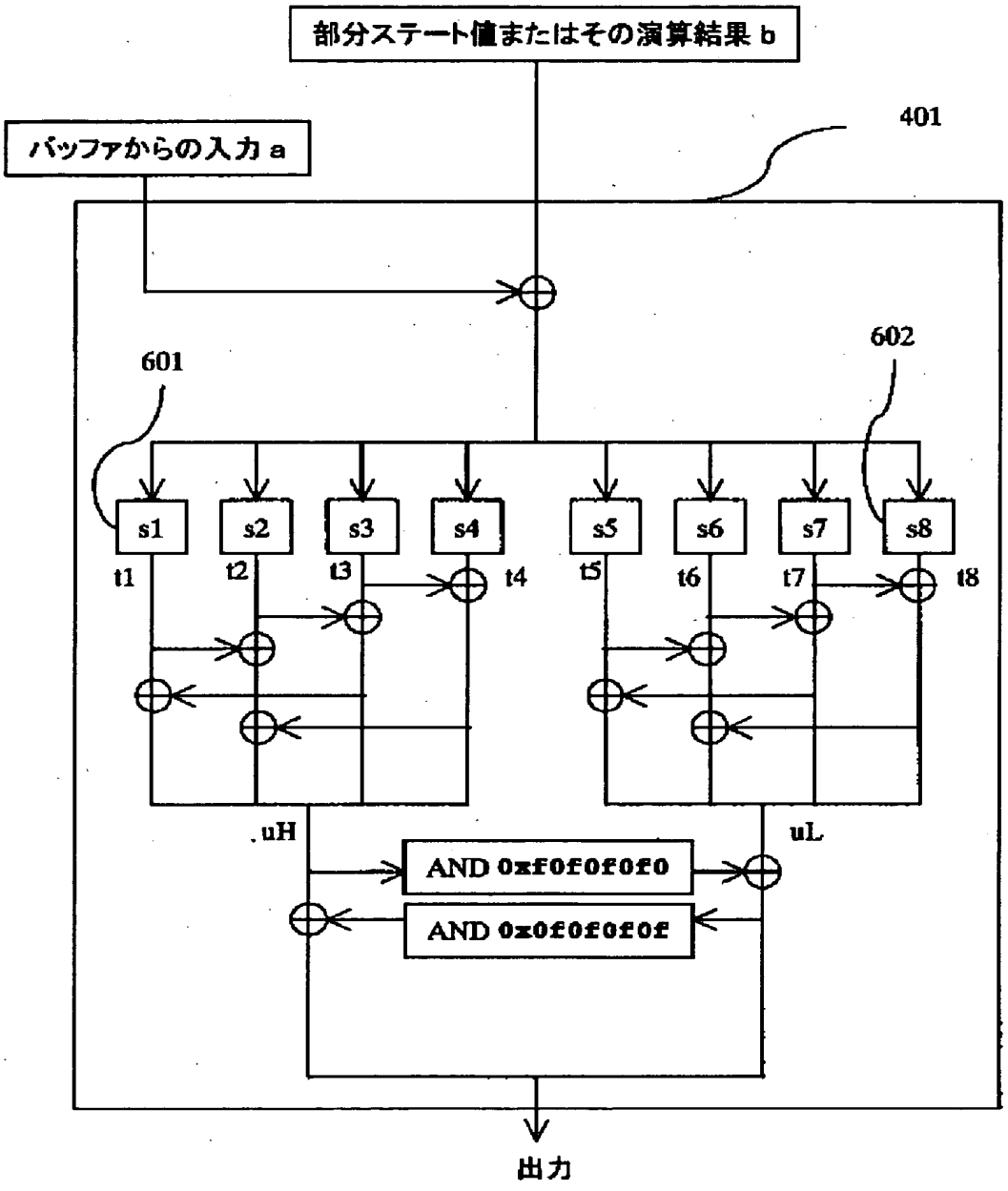
【図５】

図5



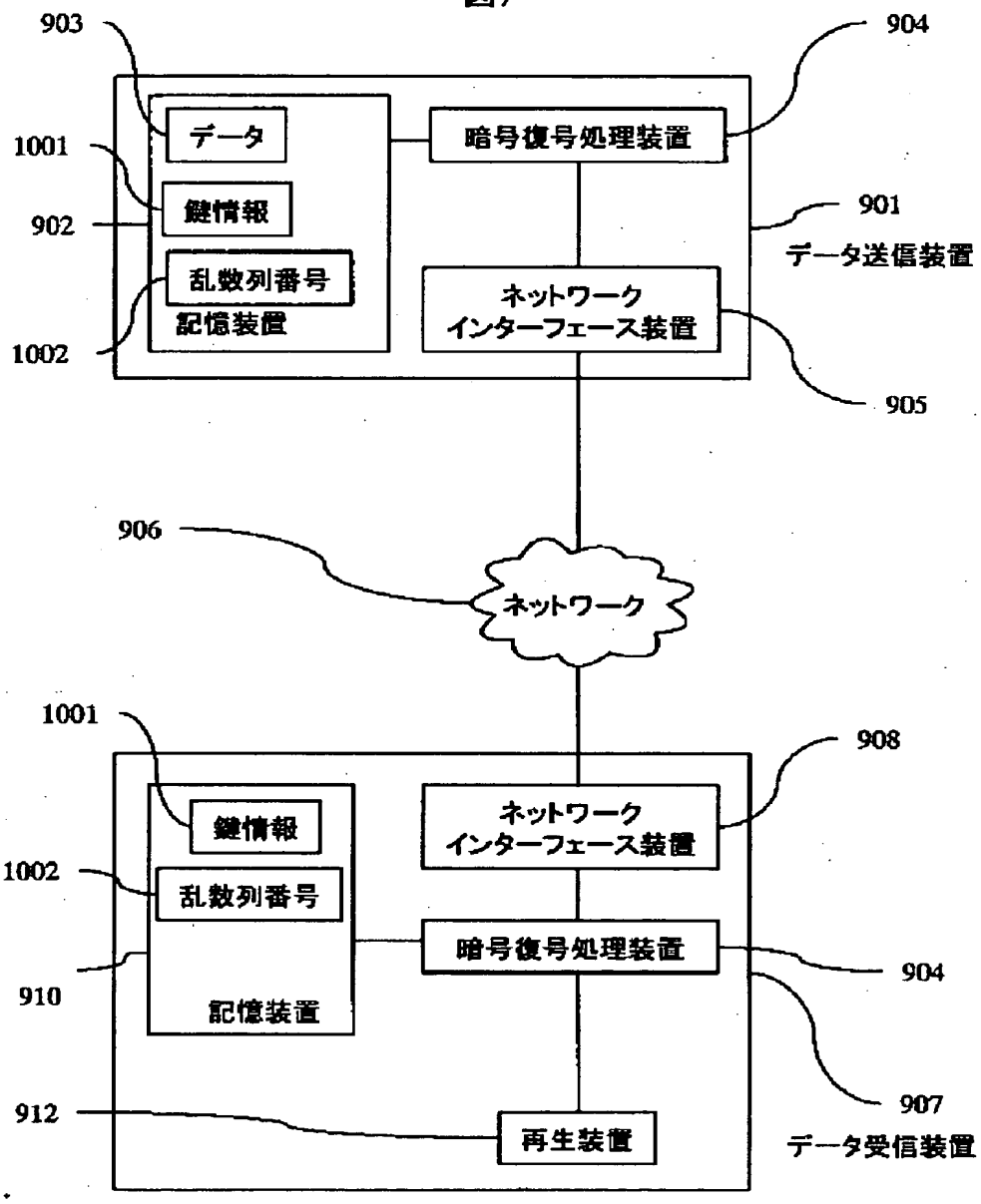
【図６】

図6


部分ステート値またはその演算結果 b
401
バッファからの入力 a
601
602
s1
s2
s3
s4
s5
s6
s7
s8
t1
t2
t3
t4
t5
t6
t7
t8
uH
uL
AND 0xf0f0f0f0
AND 0x0f0f0f0f
出力

【図７】

図7


903
904
データ
暗号復号処理装置
1001
鍵情報
902
901
データ送信装置
乱数列番号
記憶装置
ネットワーク
インターフェース装置
1002
905
906
ネットワーク
1001
908
鍵情報
ネットワーク
インターフェース装置
1002
乱数列番号
暗号復号処理装置
904
910
記憶装置
907
912
再生装置
データ受信装置

【図８】

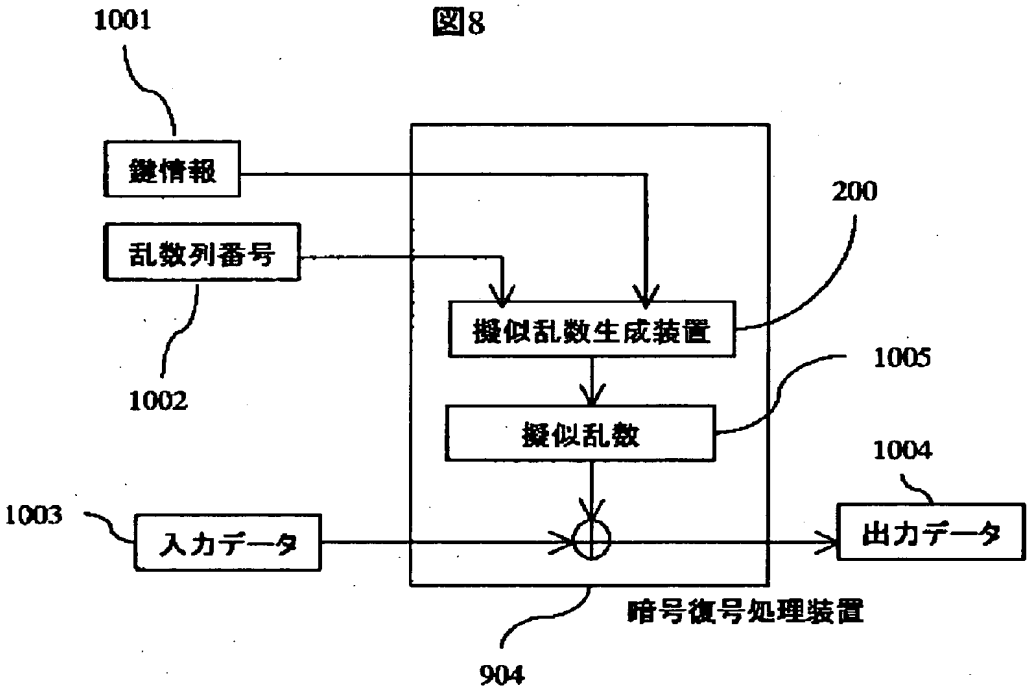
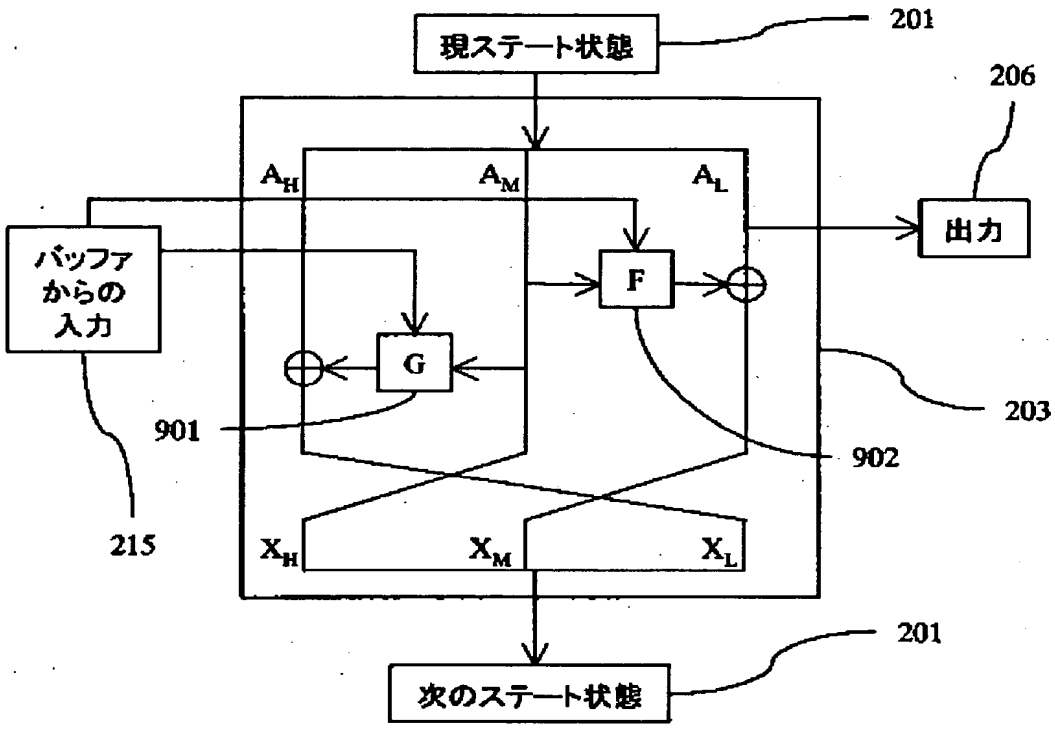

1001
鍵情報
乱数列番号
1002
200
擬似乱数生成装置
1005
擬似乱数
1004
1003
入力データ
出力データ
暗号復号処理装置
904

図8

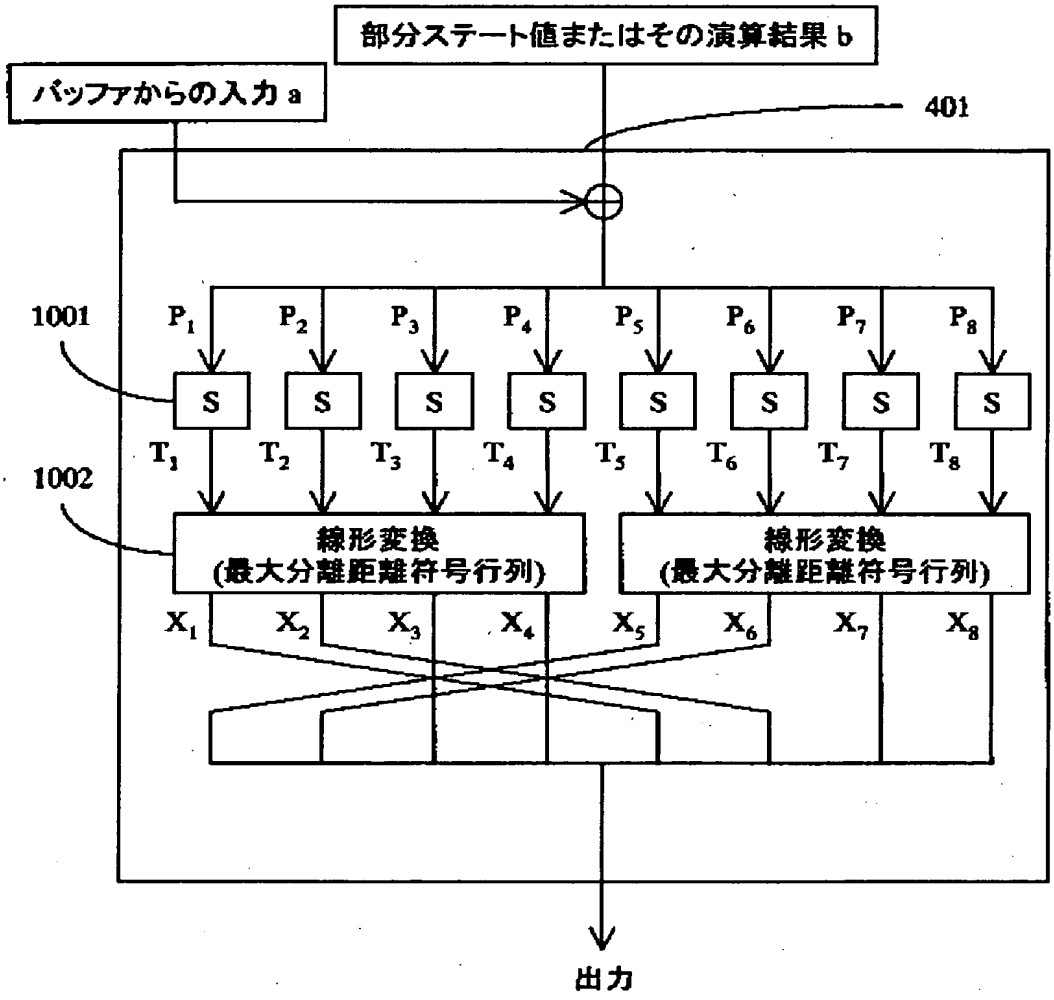
【図９】

図9


現ステート状態
201
206
A_H
A_M
A_L
出力
バッファからの入力
F
G
901
902
203
215
X_H
X_M
X_L
201
次のステート状態

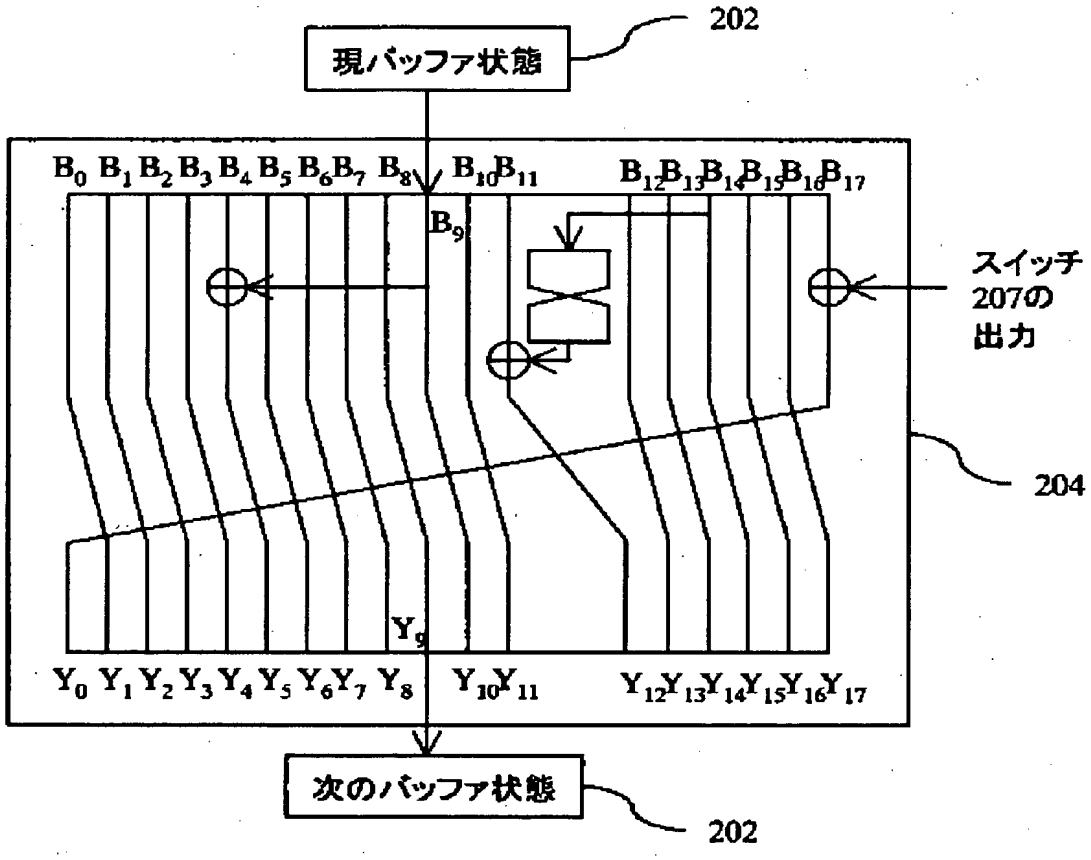
【図１０】

図10


部分ステート値またはその演算結果 b
バッファからの入力 a
401
1001
P1 P2 P3 P4 P5 P6 P7 P8
S S S S S S S S
T1 T2 T3 T4 T5 T6 T7 T8
1002
線形変換
(最大分離距離符号行列)
線形変換
(最大分離距離符号行列)
X1 X2 X3 X4 X5 X6 X7 X8
出力

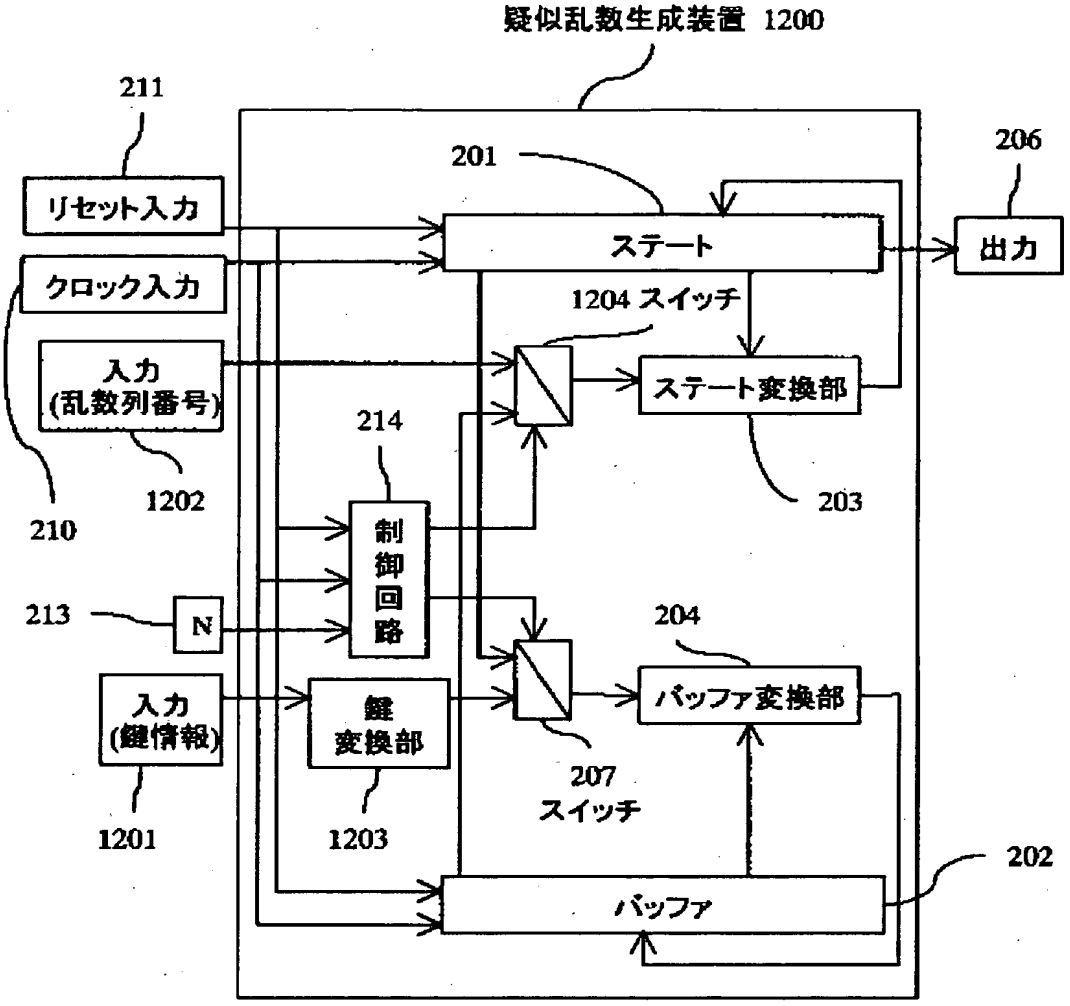
【図１１】

図11


202
現バッファ状態
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
スイッチ
207の
出力
204
Y9
Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17
次のバッファ状態
202

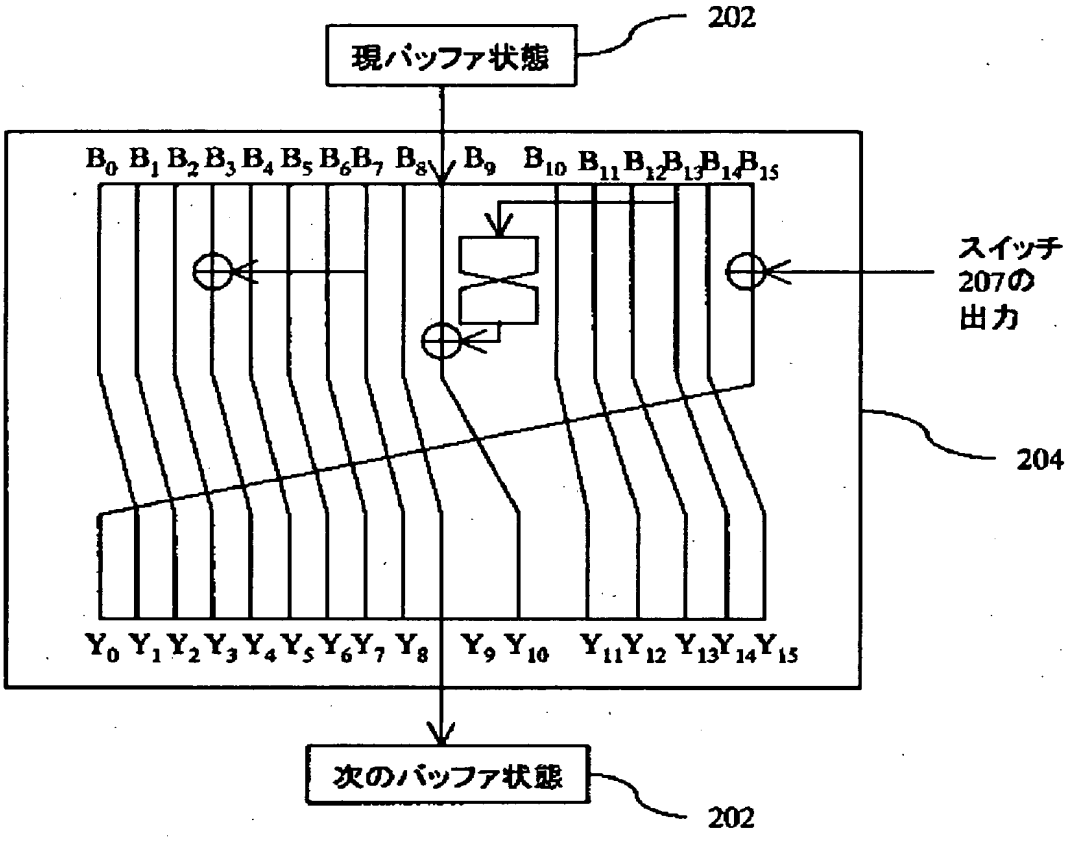
【図１２】

図12


疑似乱数生成装置 1200
211
リセット入力
クロック入力
入力
(乱数列番号)
1202
210
213
N
入力
(鍵情報)
1201
201
ステート
206
出力
1204 スイッチ
ステート変換部
203
214
制御回路
204
バッファ変換部
鍵
変換部
1203
207
スイッチ
202
バッファ

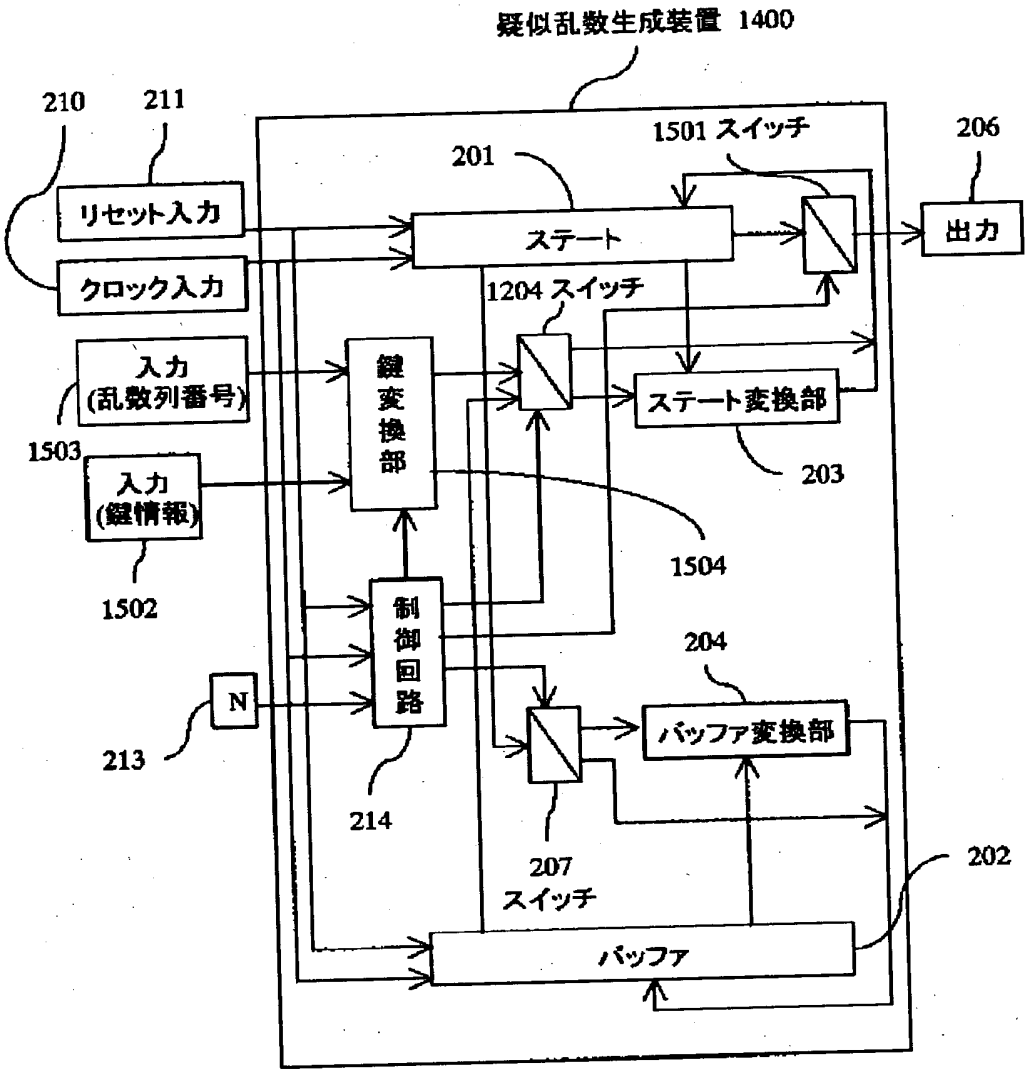
【図１３】

図13


202
現バッファ状態
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
スイッチ
207の
出力
204
Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 Y14 Y15
次のバッファ状態
202

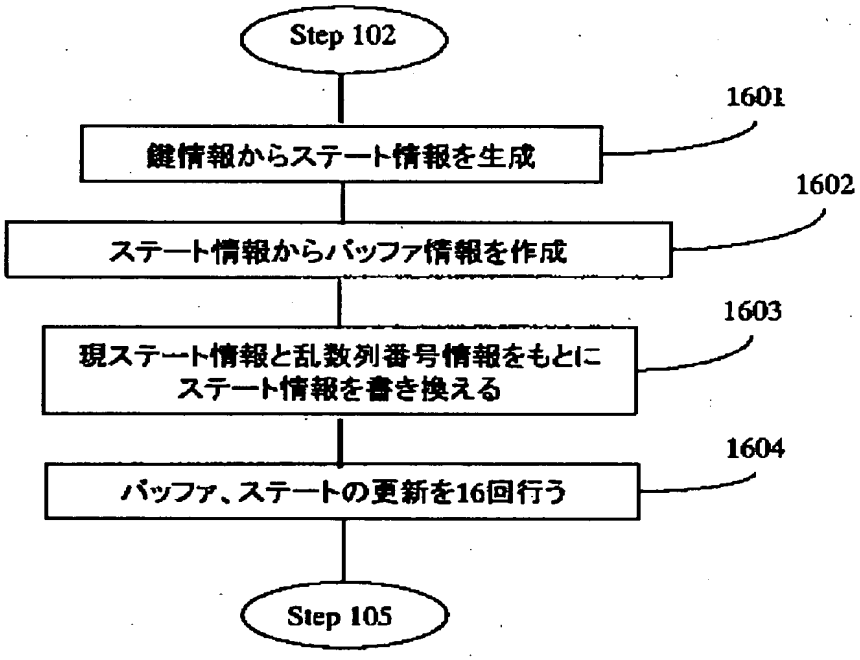
【図１４】

図14


疑似乱数生成装置 1400
210
211
リセット入力
クロック入力
201
ステート
1501 スイッチ
206
出力
1204 スイッチ
入力
(乱数列番号)
1503
入力
(鍵情報)
1502
鍵
変
換
部
ステート変換部
203
1504
制
御
回
路
N
213
214
204
バッファ変換部
207
スイッチ
202
バッファ

【図１５】

図15


Step 102
1601
鍵情報からステート情報を生成
1602
ステート情報からバッファ情報を作成
1603
現ステート情報と乱数列番号情報をもとに
ステート情報を書き換える
1604
バッファ、ステートの更新を16回行う
Step 105

【書類名】　要約書

【要約】

【課題】

ソフトウェアによる高速実行可能であり、かつ、ハードウェアでの現実的な実装が可能な疑似乱数を生成する装置を提供する。

【解決手段】

疑似乱数生成装置が備えるバッファとステートのうち、ステートについて、(i)データ処理の単位長をnとしたとき、ステートの大きさが3×nビットであり、バッファ容量が32×nビットであり、(ii)クロック制御により、時刻tから時刻t+1へのステートの状態変更を行うためのステート変換部(ステート変換関数)が、(nビット入力、nビット出力)の非線形関数Fを2回、または異なる二つの非線形関数F，Gを一回ずつ使う、という構成を提供する。ステート変換部は、暗号学的安全性と実装について十分に評価されたブロック暗号の段関数などの非線形関数の採用が可能な構成を備える。

【選択図】　図３

認定・付加情報

特許出願の番号	特願２００１－２７４４３３
受付番号	５０１０１３３１６１４
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１３年　９月１４日

＜認定情報・付加情報＞

【提出日】　平成13年 9月11日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所